

Controlling Machines from Virtual Game Through Mobile App using IoT

G. Archana
P.G Student,
MGIT Hyderabad, India

Dr. K. Sudhakar Reddy
Professor & HOD,
MGIT Hyderabad, India

Abstract— The project is designed for controlling the devices in the industry with the feedback using IoT. The feedback can be seen in the virtual app so that the users can understand the status of the devices. When the users want to control some device then they should press the button inside the app so that the device will work accordingly in the real world and can be seen as a feedback in the virtual app. The control used here is done in the two-way communication where the workers can operate the controls manually with switches or using the Android Application based on IoT technology so that the controlling can be done from any part of the world with multiple operators.

In this project we build the real-world models by importing the AutoCAD models and 3D objects and code them with operations according to the process. Then we use Raspberry Pi as a microprocessor for controlling the devices wirelessly. And we use some PHP files and host them into the server for storing the data and manipulating them. The relays are used for turning ON and OFF of the devices.

Keywords— *IoT, Virtual app, AutoCAD, Raspberry Pi, Relays and Switches.*

1. INTRODUCTION

The project aims in automating home appliances with android game using internet. Here the server and internet acts as wireless medium for controlling the home from anywhere in the world.

This project also concentrates on home automation through IoT. This system is very useful mainly in apartments, houses, malls and Industries as this eliminates the need for manual maintenance. Here the web acts as a medium to communicate. The communication range of internet is infinite and hence we get to control the devices from anywhere in the world. The internet enabled microprocessor collects the data from the cloud as the data is sent from the android game.

This project consists of internet enabled Microprocessors like Raspberry Pi, Node MCU, Analog to digital converter, potentiometer, cloud service, jumper wires, lights and motors. The android game sends the user data to the cloud and the data is received by the microprocessor which is sent to the converter which sends the signal to the appliances

1.1. *The objectives of the project include:*

- Home automation through android game.
- Wireless transmission using internet (IoT).
- Operate from anywhere in the world with Internet.

- Reduction of manual maintenance
- Usage in homes, industries, malls, apartments etc.

1.2. *Project Overview*

The project “HOME AUTOMATION THROUGH ANDROID GAME USING IoT” is developed using Unity3D game engine, Raspberry Pi and Azure cloud services. Here we control the appliances and electronics in a home which are connected to microprocessor through an android game. If we operate the appliances in the game the respective data is collected and sent to the cloud services, this data is accessed by the microprocessor and the appropriate signals are sent to the appliances as to match the game. This is done with the help of concept of Internet of Things (IoT). This application can be seen where we need to minimize the manual monitoring of the place.

2. LITERATURE SURVEY

In general, we use people to perform simple tasks like switching off and on the electrical appliances. But in here we use IoT to eliminate the physical work by connecting all the appliances to the internet and controlling them remotely through internet to perform the works for days continuously at a time without suffering from fatigue.

Raspberry Pi is a small sized or credit card sized Central Processing Unit which can connect to internet and Wi-Fi and act as an Internet of things IoT device. The advantage of Raspberry Pi is that, it is simple and easy to connect and understand. The operating system used for this device is also very user friendly and the User interference is also good and easy to understand. Raspbian OS is installed in the Raspberry Pi as the communication between the default hardware and software. Node MCU is also the type of controller. This is also a Wi-Fi enabled Internet of things (IoT) device. This device gets connected to the Wi-Fi when it gets started and shares the data through internet. Wi-Fi is the short form of wireless fidelity. It is the wireless technology as it uses the Radio Frequency to send and receive data in air at the frequency band of 2.4 GHz. The advantage of the Node MCU is that it is the open source IoT platform, low in cost, integrated support for wireless fidelity, small size of the board, and very low power consumption.

Internet which is generally called as Net in short is a worldwide system of computers or a network of networks where the user can access other by permissions to access or

get information. The devices or things which exchange their data with each other using the internet are called as Internet of Things which is shortly called as IoT. The concept of Internet of things was coined by a member of Radio frequency Identification community in the year 1999. It recently become more popular with practical mobile growth usage in the world. Internet of things are mainly categorized as internet of three things they are People to people, People to machine /things, Things /machine to things /machine, Interacting through internet. There are also few fundamental characteristics of IoT which provides Interconnectivity with global information and communication infrastructure, Things related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things, Heterogeneity as based on different platforms and networks where this can interact with other devices or service platforms, Dynamic changes, Enormous device connections to internet, safety and connectivity are the few fundamental characteristics of IoT. IoT is expected to spread rapidly over the years and this improves the quality of life of consumer and productivity of enterprises, unlocking an opportunity that which is referred as the 'Connected Life'. For consumers, the Internet of Things has the potential to deliver solutions that improves energy efficiency, education, security, health and many other aspects of human life where as for enterprises, Internet of Things can strengthen solutions to improve decision makings and productivity in manufacturing's, retails, Pharmacy, medicines, agricultures and many other sectors.

Unity3D is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as an OS X-exclusive game engine. As of 2018, the engine has been extended to support 27 platforms

Unity gives users the ability to create games in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality. Prior to C# being the primary programming language used for the engine, it previously supported Boo, which was removed in the Unity 5 release, and a version of JavaScript called Unity Script, which was deprecated in August 2017 after the release of Unity 2017.1 in favour of C#

3. METHODOLOGY

"HOME AUTOMATION THROUGH ANDROID GAME USING IoT" is developed using Unity3D game engine, Raspberry PI and Azure cloud services. Here we also used C# and Python code script to the back-end services and android user interface(front-end) is built using unity engine that control the appliances and electronics in a home which are connected to microprocessor in the game. If we operate the appliances in the game the respective data is collected and sent to the cloud services, this data is accessed by the microprocessor and the appropriate signals are sent to the appliances as to match the game. This is done with the help of concept of Internet of Things (IoT). This application can

be seen where we need to minimize the manual monitoring of the place.

3.1 IoT Devices

The Internet of Things (IoT) is a system where the transfer of data takes place between physical things embedded with sensors, software, and electronics using Internet. The sensing devices that are either passive (sensing) or active (actuating). Here the passive sensing devices are Potentiometers and the active devices are Servo motors. Here the IoT device, which makes the internet connection between the cloud and the devices are Raspberry Pi, Node MCU which are Wi-Fi internet connection enabled devices.

3.2 Unity3D

Unity3D is a platform independent game development software, we can develop both 2D and 3D games using this software. C# scripting is used for the functionality of the game. We use JDK (java development kit)8.1 and Android SDK (software development kit) to build the android application

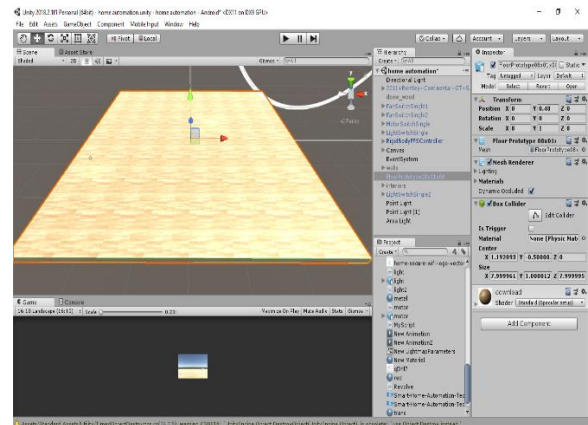


Figure 3.1 Interface

As we see in the Figure 3.1, we can see the interface of Unity3D, this has 6 tools that help to build, place, scale, rotate any object in the large 3D space available in the software. Inspector tab provides all the information about the orientation of the objects or the 3D models in the large 3D space

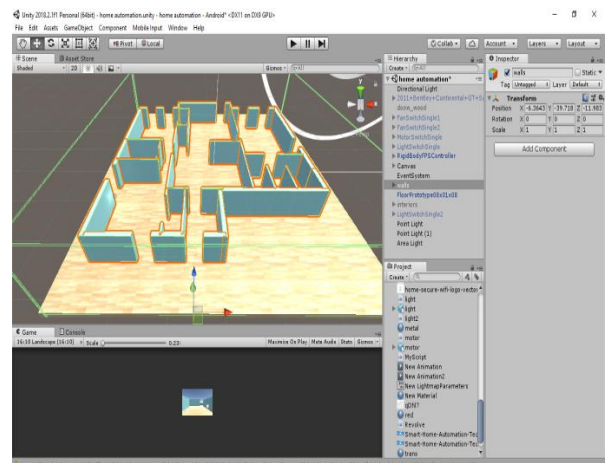


Figure 3.2 Design

In the figure 3.2 (design) we can see the layout of the house being constructed virtually in the game engine using unity tools.

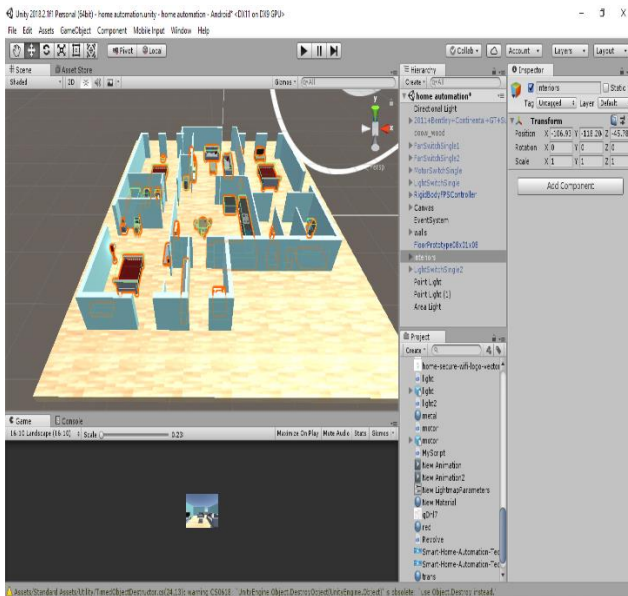


Figure 3.3

In this figure 3.3, you can see the 3D furniture in the house being virtually imported. The hierarchy tab in the interface displays all the 3D models used in the project, this tab also provides information about the assets in the project like main camera and lighting in the game.

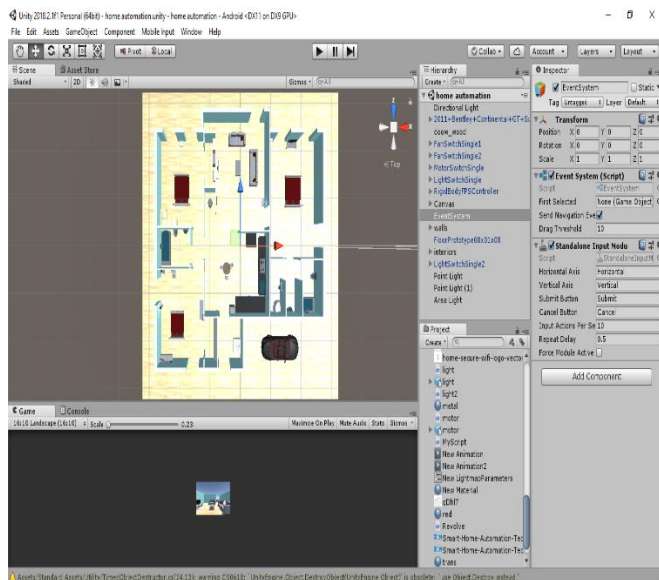


Figure 3.4

In the figure 3.4 we can see the top view of the layout in such a way that we see the arrangement of the 3D layout. Unity also provides new materials that we want to add to the 3d objects.

We can add the textures, materials and colors to the 3d model. We can adjust the orientation of the 3d model using the transforms tab in inspector tab

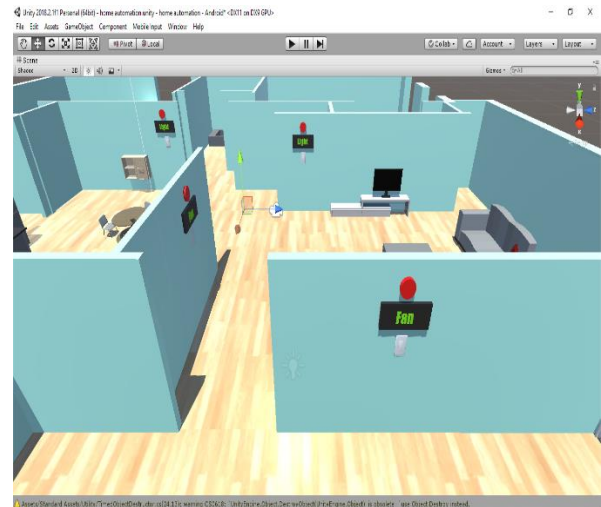


Figure 3.5

Buttons are placed in the project using C# code script by generating new events in the project. This button is interactive in the game, which later will be attached to the cloud service platform and then connected to the Raspberry PI which interacts with the electrical appliances in the home.

Colliders are used in the interface to call the hyperlinks in the back-end.

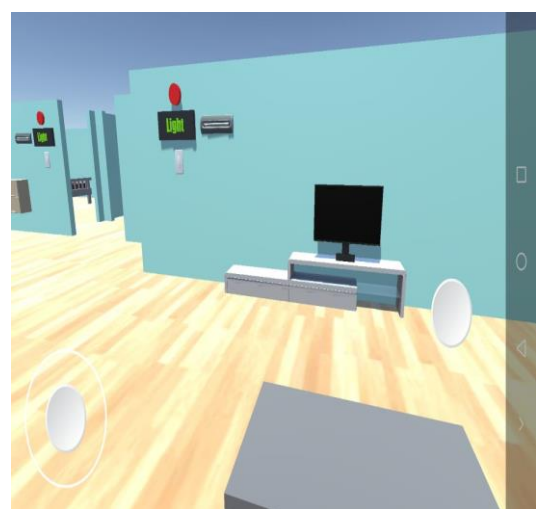


Figure 3.6

The figure 3.6 is the interface of the game, the main camera acts as the first-person perspective. The player can interact the game using the joystick in the UI to the left bottom that helps to move and look around in the game.

The other button to the left is to jump in the game.

3.3 Raspberry Pi

The Raspberry Pi is the tiny PC which is in the size of a credit card. The Raspberry Pi was created in February 2012. It was founded by the Raspberry Pi foundation in 2009. The Raspberry Pi foundation is the charity-based foundation in United Kingdom which enables children and people of all age groups to explore computing, projects and do programming in simple programming language like Python. The Raspberry Pi is being used by the children and all age groups all over the world to learn programming and understand how computers work. The word Raspberry Pi is the mixture of two words Raspberry and Pi as shown in the Figure 3.19.

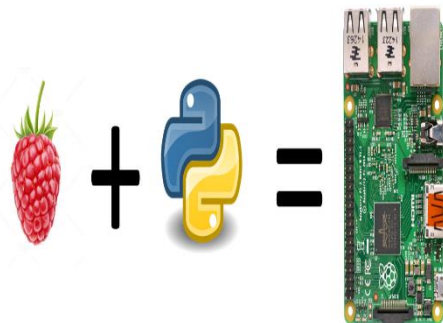


Figure 3.7 Raspberry and Python

In those there is a tradition for naming after the fruits names as they will be a cache word for people to remember. So, the Raspberry is the most popular fruit in United Kingdom and Python is the recommended language for it. So, the mixture of them they named it as Raspberry Pi. The Raspberry Pi consists of all the elements that a Central Processing Unit has in it as shown in the Figure 3.20 below.

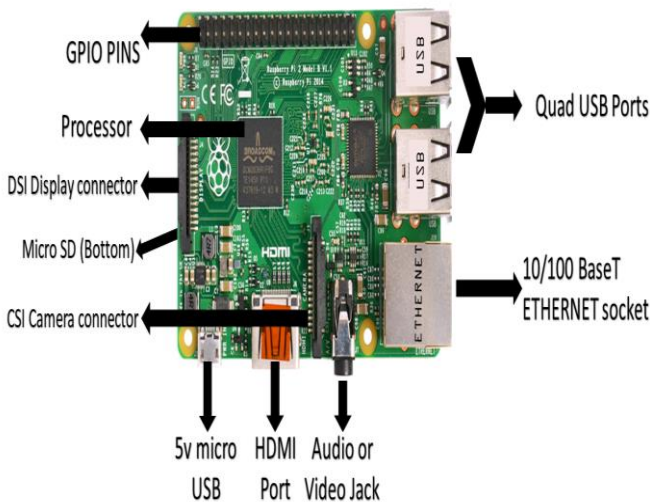


Figure 3.8 Raspberry Pi

1. Ethernet Socket for connecting Local Area Network Cable for internet connection.
2. Stereo Audio/Video Jack for stereo output Audio or Video.

3. High Definition Multimedia Interface port for making the connection between the Raspberry Pi and Desktop.
4. Five volts Micro Universal Serial Bus port for input power supply.
5. Camera Serial Interface Camera port for connecting Camera.
6. Micro SD card which is situated below the DSI Display connector is used for inserting the Micro SD card into it which has the operating system in it.
7. Four Universal Serial Bus ports so that Keyboard, Mouse, Pen drives etc... Can be connected to them.
8. Processor used here is Broadcom BCM2837 64 Bit Quad Core CPU at 1.2GHz, and 1Gigabytes of Random-access memory.
9. There are 40 pins where some are Vcc pins, Ground pins, do not connect pins, Rx pins, Tx pins, and 26 General purpose Input and Output pins.

3.4 Installing Operating System in Raspberry Pi

The operating system is the basic system software which acts as the interference between the default hardware and software (or) the operating system is the low-level software that supports computers basic functions. There are many different operating systems like Windows, MacOS, Android, Linux etc... The Raspbian is the special Operating System which is built for Raspberry Pi. The name "Raspbian" Operating system is derived by the mixture of two words i.e., Raspberry Pi + Debian. Debian is the one of the stable Linux Operating system. First download the Raspbian Operating system from the Raspberry Pi official website will be as shown in the Figure 3.21 and select the Raspbian Operating system and download the file using zip or torrent file_www.raspberrypi.org/downloads

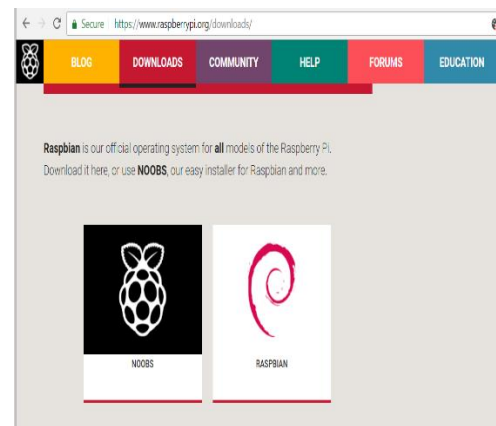


Figure 3.9 Raspbian download

Then similarly download the SD formatter and win32 disc imager from the tags below:

SD card formatter
Win32 disk imager

Install the applications after download. There are three steps to be followed during the booting process they are:

1. Insert the SD card in the Memory Card reader and connect it to PC. So, the connection is established between the storage drive and PC.

- Then Open the SD formatter as in Figure 3.22 → Select the drive → Format. In this step old data will be formatted and the drive will be ready for the process.
 Note: Please select the drive properly as it formats the data

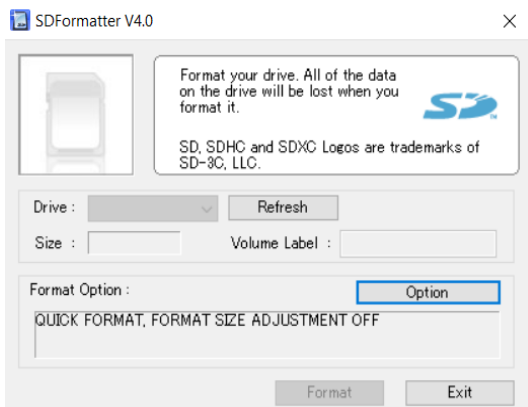


Figure 3.10 SD Formatter

- Then open the Win32 Disk imager as in Figure 3.23. Then browse for the Raspbian.img file and select it. Click on WRITE, then a pop-up window opens and asks for the attention then click ok. The process starts immediately and reaches till 100% which means Booting process is completed.

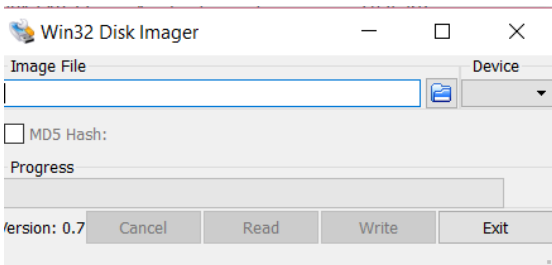


Figure 3.10 Win32 Disk Imager

3.5 Programming in Raspberry Pi

After the booting process insert the memory card into Raspberry Pi and connect HDMI cable from Raspberry Pi to Desktop, keyboard, mouse to Raspberry Pi and finally connect the power supply and turn it ON. Then the Raspberry Pi works as the PC and is ready to use. Open the Terminal and enter following commands:

→ cd Desktop

Then the path changes to the Desktop folder.

→ touch ledblink.py

This touch command will create a new empty file on Desktop.

→ sudo nano ledblink.py

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(12,GPIO.OUT)

while True:

    print ("on")
    GPIO.output(12,1)
    time.sleep(2)
    print ("off")
    GPIO.output(12,0)
    time.sleep(2)

GPIO.cleanup()
```

Then editor will open to write code below code should be typed in to the editor

→CTRL+X

→Y

→ Enter

The program will be saved into the file ledblink.py. The file can also be accessed from the desktop also. Then connect a Light Emitting Diode with 220 ohms and connect the Light Emitting Diode Anode to pin 12 and cathode pin to Ground i.e., pin 6 as shown in the Figure 3.24. Then run the following command to run the program.

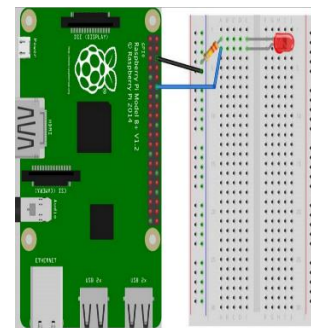


Figure 3.24 LED to Raspberry Pi

→ sudo python ledblink.py

Then the Light Emitting Diode should be blinking ON for 2 seconds and OFF for 2 seconds continuously.

3.6 Python

Python is a General-Purpose programming language for interpreted high-level programming language. The logo of Python is shown in the Figure 3.25. The Python language was created by Guido Van Rossum and it was released firstly in the year 1991. Now in 2018 the Python language stands among the top five powerful programming language in the world. Python is the powerful interpreted object-oriented, high-level language. The python has some feature like it is simple so it is also used by the most of kids all over the world as it also has easy to use syntax and the source code is also available under the General Public License (GPL). The python language is widely used because of its complexity is simple and the coding can be done in few lines when compared to other programming languages.



Figure 3.11 Python

3.7 AZURE

The Azure hosts the websites using the web frame. Built-in GUI helps us to create a server to build a communication between the devices.

3.8 Internet of Things (IoT)

Internet of Things (IoT) is network of electronic devices, physical devices, home appliances, sensors connected to devices, actuators, medical devices, etc... Which enables these devices and sensors send and receive data with each other. The IoT is the inter linking of devices like sensors, devices, software, electronics that enable the devices to collect and transfer data. Internet of things is also referred as the Internet of Everything (IoE) which also consists of all the web enabled devices to exchange their data. Internet of things has evolved due to the convergence of multiple technologies, machine learning, artificial intelligence, real time analysis, sensors and embedded systems. Traditional methods of embedded systems, wireless networks of sensors, control systems etc. Contribute to enabling of Internet of Things.

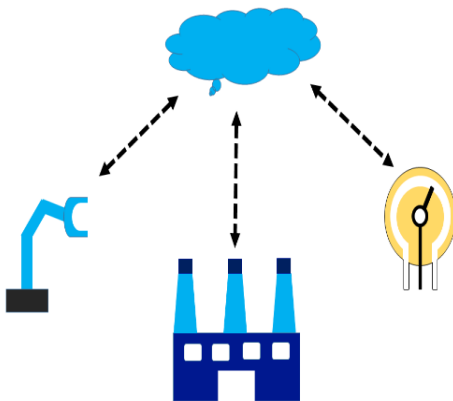


Figure 3.12 IoT Network

The devices are connected to Internet access data via web services to perform specific task.

4. C# SCRIPTING

4.1 Script for fan

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.Networking;
```

```
public class fan : MonoBehaviour  
{  
    private string UpdateOnURL =  
    "http://ky.azurewebsites.net/devcall?device=mtechproject&  
    dev_call=s31";  
    private string UpdateOffURL =  
    "http://ky.azurewebsites.net/devcall?device=mtechproject&  
    dev_call=s30";
```

```
    public Collider coll;  
    public GameObject greenON, redOFF;  
    private bool fanON = false;
```

```
    private void Awake()  
    {  
        coll = GetComponent<Collider>();  
    }
```

```
    public void OnMouseDown()  
    {  
        fanON = !fanON;
```

```
        coll.enabled = false;
```

```
        if ( fanON == true )  
        {  
            StartCoroutine(FanOn());  
        }
```

```
        else if(fanON == false)
```

```
        {  
            string UpdateOff_URL = this.UpdateOffURL;  
            WWW loginReader = new  
            WWW(UpdateOff_URL);  
            StartCoroutine(FanOff());  
        }
```

```
    private IEnumerator FanOn()  
    {
```

```
        yield return new WaitForSeconds(0.1f);  
        greenON.SetActive(true);  
        redOFF.SetActive(false);  
        string UpdateOn_URL = this.UpdateOnURL;  
        WWW loginReader = new WWW(UpdateOn_URL);  
        coll.enabled = true;
```

```
}  
  
private IEnumerator FanOff()  
{  
    yield return new WaitForSeconds(0.1f);  
    greenON.SetActive(false);  
    redOFF.SetActive(true);  
    coll.enabled = true;  
}  
}
```

4.2 Script for fixed buttons

```
using UnityEngine;  
using UnityEngine.EventSystems;  
  
public class FixedButton : MonoBehaviour,  
IPointerUpHandler, IPointerDownHandler  
{  
    [HideInInspector]  
    public bool Pressed;  
  
    // Use this for initialization  
    void Start()  
    {  
    }  
  
    // Update is called once per frame  
    void Update()  
    {  
    }  
  
    public void OnPointerDown(PointerEventData  
eventData)  
    {  
        Pressed = true;  
    }  
  
    public void OnPointerUp(PointerEventData eventData)  
    {  
        Pressed = false;  
    }  
}
```

4.3 Script for Fixed touch field

```
using UnityEngine;  
using UnityEngine.EventSystems;  
  
public class FixedTouchField : MonoBehaviour,  
IPointerDownHandler, IPointerUpHandler  
{  
    [HideInInspector]  
    public Vector2 TouchDist;  
    [HideInInspector]  
    public Vector2 PointerOld;  
    [HideInInspector]  
    protected int PointerId;  
    [HideInInspector]
```

```
public bool Pressed;  
  
    // Use this for initialization  
    void Start()  
    {  
    }  
  
    // Update is called once per frame  
    void Update()  
    {  
        if (Pressed)  
        {  
            if (PointerId >= 0 && PointerId <  
Input.touches.Length)  
            {  
                TouchDist = Input.touches[PointerId].position -  
PointerOld;  
                PointerOld = Input.touches[PointerId].position;  
            }  
            else  
            {  
                TouchDist = new Vector2(Input.mousePosition.x,  
Input.mousePosition.y) - PointerOld;  
                PointerOld = Input.mousePosition;  
            }  
        }  
        else  
        {  
            TouchDist = new Vector2();  
        }  
    }  
  
    public void OnPointerDown(PointerEventData  
eventData)  
    {  
        Pressed = true;  
        PointerId = eventData.pointerId;  
        PointerOld = eventData.position;  
    }  
  
    public void OnPointerUp(PointerEventData eventData)  
    {  
        Pressed = false;  
    }  
}
```

4.4 script for lights

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.Networking;  
  
public class light : MonoBehaviour  
{  
    private string UpdateOnURL =  
"http://ky.azurewebsites.net/devcall?device=mtechproject&  
dev_call=s21";
```

```
private string UpdateOffURL =
"http://ky.azurewebsites.net/devcall?device=mtechproject&
dev_call=s20";

public Collider col4;
public GameObject greenON, redOFF, light1ON;
private bool lightON = false;

private void Awake()
{
    col4 = GetComponent<Collider>();
}

public void OnMouseDown()
{
    lightON = !lightON;

    col4.enabled = false;

    if (lightON == true)
    {
        StartCoroutine(LightOn());
    }
    else if (lightON == false)
    {
        string UpdateOff_URL = this.UpdateOffURL;
        WWW loginReader = new
WWW(UpdateOff_URL);
        StartCoroutine(LightOff());
    }
}

private IEnumerator LightOn()
{
    yield return new WaitForSeconds(0.1f);
    greenON.SetActive(true);
    light1ON.SetActive(true);
    redOFF.SetActive(false);
    string UpdateOn_URL = this.UpdateOnURL;
    WWW loginReader = new WWW(UpdateOn_URL);
    col4.enabled = true;
}

private IEnumerator LightOff()
{
    yield return new WaitForSeconds(0.1f);
    greenON.SetActive(false);
    light1ON.SetActive(false);
    redOFF.SetActive(true);
    col4.enabled = true;
}
}
```

4.5 motor

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
using UnityEngine.Networking;

public class motor : MonoBehaviour
{
    private string UpdateOnURL =
"http://ky.azurewebsites.net/devcall?device=mtechproject&
dev_call=s11";
    private string UpdateOffURL =
"http://ky.azurewebsites.net/devcall?device=mtechproject&
dev_call=s10";

    public Collider col3;
    public GameObject greenON, redOFF;
    private bool motorON = false;

    private void Awake()
    {
        col3 = GetComponent<Collider>();
    }

    public void OnMouseDown()
    {
        motorON = !motorON;

        col3.enabled = false;

        if (motorON == true)
        {
            StartCoroutine(MotorOn());
        }
        else if (motorON == false)
        {
            string UpdateOff_URL = this.UpdateOffURL;
            WWW loginReader = new
WWW(UpdateOff_URL);
            StartCoroutine(MotorOff());
        }
    }

    private IEnumerator MotorOn()
    {
        yield return new WaitForSeconds(0.1f);
        greenON.SetActive(true);
        redOFF.SetActive(false);
        string UpdateOn_URL = this.UpdateOnURL;
        WWW loginReader = new WWW(UpdateOn_URL);
        col3.enabled = true;
    }

    private IEnumerator MotorOff()
    {
        yield return new WaitForSeconds(0.1f);
        greenON.SetActive(false);
        redOFF.SetActive(true);
        col3.enabled = true;
    }
}
```


4.6 MY SCRIPT

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.StandardAssets.Characters.FirstPerson;

public class MyScript : MonoBehaviour {
    public FixedJoystick MoveJoystick;
    public FixedButton JumpButton;
    public FixedTouchField TouchField;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        var fps = GetComponent<RigidbodyFirstPersonController>().fps;
        fps.RunAxis = MoveJoystick.inputVector;
        fps.JumpAxis = JumpButton.Pressed;
        fps.mouseLook.LookAxis = TouchField.TouchDist;
    }
}
```

4.7 revolve

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Revolve : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update ()
    {
        transform.RotateAround(Vector3.zero, Vector3.up, .3f
* Time.deltaTime);
    }
}
```

5. ADVANTAGES AND APPLICATIONS

5.1 Advantages

1. Operation of this Game is simple.
2. No need of trained professionals to operate this even untrained can also operate this easily.
3. This can be operated from anywhere in the world
4. Simplest, Cheapest and widest communication as this works using Internet.

5.2 Applications

This project has many applications in the real world as it replicates the human motion from anywhere in the world using the internet.

This can be used in any apartments, industries, factories, and many more places that completely eliminates wastage of human monitoring efforts.

6. CONCLUSION AND FUTURE SCOPE

The conclusions are drawn from the present research work of this project using IoT are as follows:

- It is the future Technology where we operate easily.
- This is very quick in action and works accurately. The further improvements of this project would be with the computer interference.

REFERENCES

- [1] G. Ashish Jadhav, Ashish Ranjan, Ankit, Aditi Chaudhuri, "Controlling a Raspberry Pi Robot over Internet with HTML & Shell Scripts", IRJET, Volume: 04, Issue: 03 Mar -2017.
- [2] Suraj Tonage, Sandhya Yemul, Rajendra Jare, Veena Patki, "IoT based home automation system using NodeMCU ESP8266 module" IJARND Volume3, Issue3 2018.
- [3] Internet of Things Publisher: Orient Black Swan ISBN 10: 8173719543 ISBN 13: 9788173719547 Publisher: Universities Press, 2015.
- [4] K Keyur Patel, M Sunil Patel, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges", IJESE, Volume 6 Issue No. 5, 2016.
- [5] Understanding the Internet of Things (IoT), 2014 GSM Association. July 2014.
- [6] S. Prasanna Bhoite, Madan B. Mali, "An Internet Controlled Electronics Using an Android Mobile ", IJARCCCE, Vol. 5, Issue 6, June 2016.